PRÁCTICA 3

Arquitectura de servicios red

Javier Álvarez Martínez
201707599@alu.comillas.edu

1ª SOLUCIÓN

1. Primero permitimos conexiones http a nuestro servicio web en el puerto 80. Conectamos las reglas de firewall a través del network tag, añadiendo el nombre de la regla de firewall.
2. Descargamos nginx(sudo apt install nginx) y llamamos a la IP pública externa de nuestro servidor

34.170.215.16

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

3. Nos conectamos mediante ssh, usando nuestra clave publica a nuestra máquina de salto.

```
C:\Users\javie>ssh 104.154.86.86
The authenticity of host '104.154.86.86 (104.154.86.86)' can't be established.
ECDSA key fingerprint is SHA256:G8EAAtt6ra4eRhQ696o4sw0VF93D2owwR1uvkEUBols.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '104.154.86.86' (ECDSA) to the list of known hosts.
Linux maquinasalto 5.10.0-17-cloud-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
javie@maquinasalto:~$
```

4. Nos conectamos a través de la máquina de salto y usando la clave pública de la máquina de salto y mediante ssh a nuestro servicio web.

```
javie@maquinasalto:~$ ssh 10.128.0.5
The authenticity of host '10.128.0.5 (10.128.0.5)' can't be established.
ECDSA key fingerprint is SHA256:mVDmJcjNxVCPo5B3XsyaI7UIqq74TQSiNxU6iMRp/zU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.128.0.5' (ECDSA) to the list of known hosts.
Linux servicioweb 5.10.0-17-cloud-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
javie@servicioweb:~$
```

5. Comprobamos como no nos deja conectarnos a nuestro servicio web a través de un equipo que no sea nuestra máquina de salto

```
C:\Users\javie>ssh 34.170.215.16
ssh: connect to host 34.170.215.16 port 22: Connection timed out
```

6. Mostramos las reglas de firewall creadas.

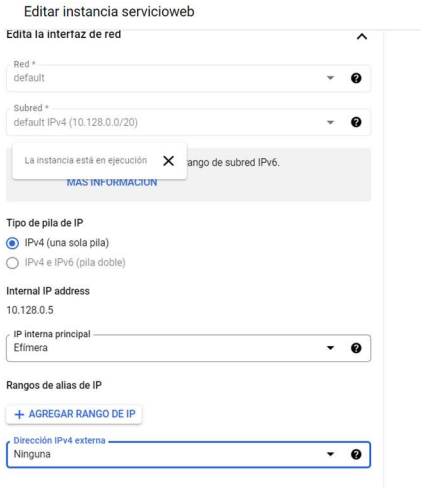
Nombre	Tipo	Destinos	Filtros	Protocolos/puertos	Acción	Prioridad	Red	Registros	Re
http-servidor	Entrada	http-servicio	Intervalos de	tcp:80	Permitir	1000	default	Desactivado	
ssh-salto	Entrada	ssh-salto	Intervalos de	tcp:22	Permitir	1000	default	Desactivado	
ssh-servidor	Entrada	ssh-server	Intervalos de	tcp:22	Permitir	1000	default	Desactivado	

7. Mostramos las máquinas virtuales

Estado	Nombre	Zona	Recomendaciones	En uso por	IP interna	IP externa	Conectar
✓	maquinasalto	us-central1-a			10.128.0.3 (nic0)	104.154.86.86 (nic0)	SSH
✓	servicioweb	us-central1-a			10.128.0.5 (nic0)	34.170.215.16 (nic0)	SSH

2ª SOLUCIÓN

1. Quitamos la ip pública del servicio web



2. Mostramos como no tiene ip externa nuestro servicio web

☐	✓	servicioweb	us-central1-a	10.128.0.5 (nic0)	SSH

3. Demostramos que no podemos conectarnos a internet al quitar la IP pública ya que no podemos descargar nginx.

```
javie@servicioweb:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libnginx-mod-stream-geoip nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libnginx-mod-stream-geoip nginx nginx-common nginx-core
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 1416 kB of archives.
After this operation, 2818 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Err:1 http://deb.debian.org/debian bullseye/main amd64 nginx-common all 1.18.0-6.1+deb11u2
  Could not connect to debian.map.fastlydns.net:80 (199.232.98.132), connection timed out Could not connect to deb.debian.org:80 (146.75.78.132), connection timed out
Err:2 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-http-geoip amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:3 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:4 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:5 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-mail amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:6 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-stream amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:7 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-stream-geoip amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:8 http://deb.debian.org/debian bullseye/main amd64 nginx-core amd64 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
Err:9 http://deb.debian.org/debian bullseye/main amd64 nginx all 1.18.0-6.1+deb11u2
  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/nginx-common_1.18.0-6.1%2bdeb11u2_all.deb  Could not connect to debian.map.fastlydns.net:80 (199.232.98.13
2), connection timed out Could not connect to deb.debian.org:80 (146.75.78.132), connection timed out
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-http-geoip_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-http-image-filter_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-http-xslt-filter_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-mail_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-stream_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
```

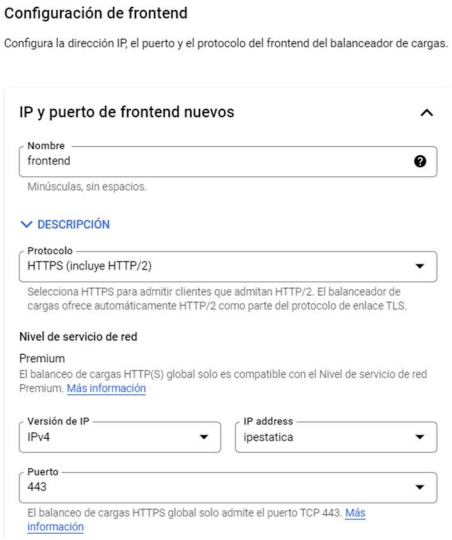
```
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/nginx-common_1.18.0-6.1%2bdeb11u2_all.deb  Could not connect to debian.map.fastlydns.net:80 (199.232.98.132), connection timed out Could not connect to deb.debian.org:80 (146.75.78.132), connection timed out
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-http-geoip_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-http-image-filter_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-http-xslt-filter_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-mail_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-stream_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/libnginx-mod-stream-geoip_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/nginx-core_1.18.0-6.1%2bdeb11u2_amd64.deb  Unable to connect to deb.debian.org:http:
E: Failed to fetch http://deb.debian.org/debian/pool/main/n/nginx/nginx_1.18.0-6.1%2bdeb11u2_all.deb  Unable to connect to deb.debian.org:http:
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
javie@servicioweb: $
```

4. Utilizamos nat para tener conectividad a internet.

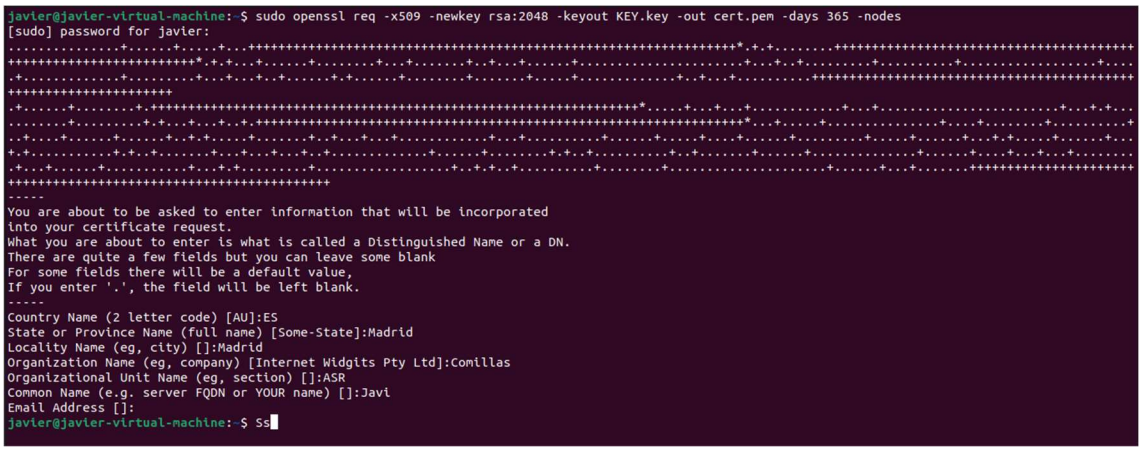
5. Vemos como ahora si podemos descargar nginx

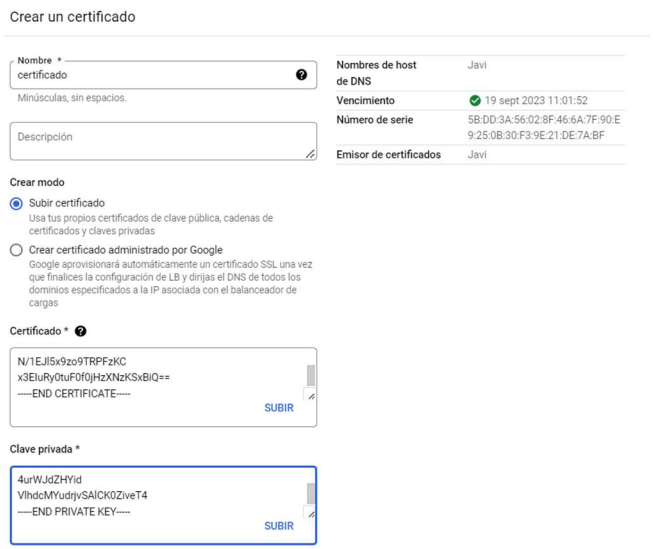
```
javie@servicioweb: $ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip nginx-common
  nginx-core
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip nginx nginx-common
  nginx-core
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 1416 kB of archives.
After this operation, 2818 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 nginx-common all 1.18.0-6.1+deb11u2 [126 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-http-geoip amd64 1.18.0-6.1+deb11u2 [98.4 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-6.1+deb11u2 [102 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-6.1+deb11u2 [100 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-mail amd64 1.18.0-6.1+deb11u2 [129 kB]
Get:6 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-stream amd64 1.18.0-6.1+deb11u2 [155 kB]
Get:7 http://deb.debian.org/debian bullseye/main amd64 libnginx-mod-stream-geoip amd64 1.18.0-6.1+deb11u2 [97.6 kB]
Get:8 http://deb.debian.org/debian bullseye/main amd64 nginx-core amd64 1.18.0-6.1+deb11u2 [515 kB]
Get:9 http://deb.debian.org/debian bullseye/main amd64 nginx all 1.18.0-6.1+deb11u2 [92.9 kB]
Fetched 1416 kB in 15s (91.6 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 53996 files and directories currently installed.)
Preparing to unpack .../0-nginx-common_1.18.0-6.1+deb11u2_all.deb ...
Unpacking nginx-common (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-http-geoip.
Preparing to unpack .../1-libnginx-mod-http-geoip_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-http-geoip (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-http-image-filter.
Preparing to unpack .../2-libnginx-mod-http-image-filter_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-http-geoip (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-http-image-filter.
Preparing to unpack .../2-libnginx-mod-http-image-filter_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-http-image-filter (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-http-xslt-filter.
Preparing to unpack .../3-libnginx-mod-http-xslt-filter_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-http-xslt-filter (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-mail.
Preparing to unpack .../4-libnginx-mod-mail_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-mail (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-stream.
Preparing to unpack .../5-libnginx-mod-stream_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-stream (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package libnginx-mod-stream-geoip.
Preparing to unpack .../6-libnginx-mod-stream-geoip_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking libnginx-mod-stream-geoip (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package nginx-core.
Preparing to unpack .../7-nginx-core_1.18.0-6.1+deb11u2_amd64.deb ...
Unpacking nginx-core (1.18.0-6.1+deb11u2) ...
Selecting previously unselected package nginx.
Preparing to unpack .../8-nginx_1.18.0-6.1+deb11u2_all.deb ...
Unpacking nginx (1.18.0-6.1+deb11u2) ...
Setting up nginx-common (1.18.0-6.1+deb11u2) ...
Setting up libnginx-mod-http-xslt-filter (1.18.0-6.1+deb11u2) ...
Setting up libnginx-mod-http-geoip (1.18.0-6.1+deb11u2) ...
Setting up libnginx-mod-mail (1.18.0-6.1+deb11u2) ...
Setting up libnginx-mod-http-image-filter (1.18.0-6.1+deb11u2) ...
Setting up libnginx-mod-stream (1.18.0-6.1+deb11u2) ...
Setting up libnginx-mod-stream-geoip (1.18.0-6.1+deb11u2) ...
Setting up nginx-core (1.18.0-6.1+deb11u2) ...
Upgrading binary: nginx.
Setting up nginx (1.18.0-6.1+deb11u2) ...
Processing triggers for man-db (2.9.4-2) ...
```

6. Creamos balanceador de cargas. Primeramente, debemos configurar el frontend de nuestro balanceador, queremos utilizar un protocolo HTTPS y con una ip estática.

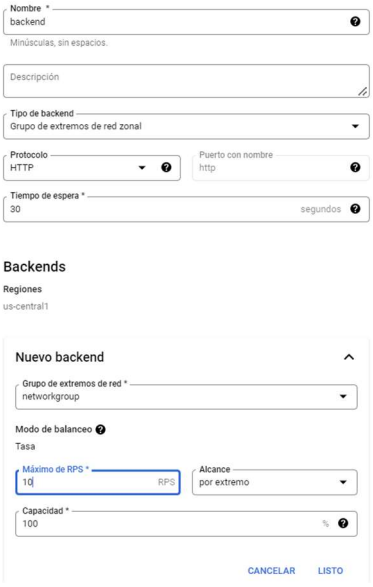


A continuación, debemos crearnos un certificado con el objetivo de que sea lo más fiable posible permitiendo https.

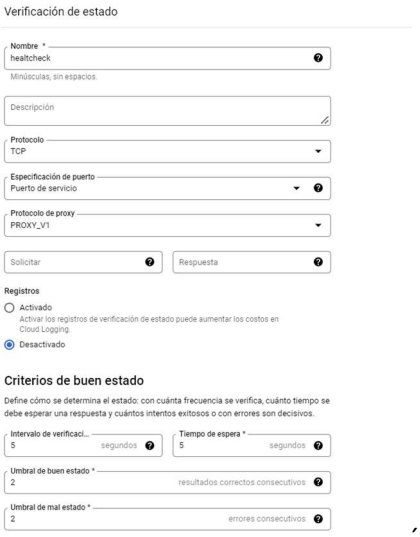




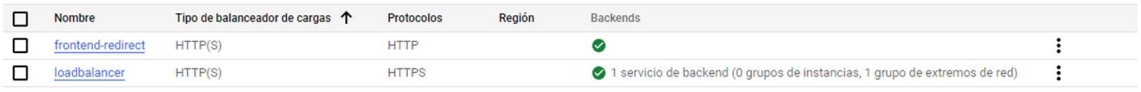
A continuación, configuramos el backend de nuestro loadbalancer.



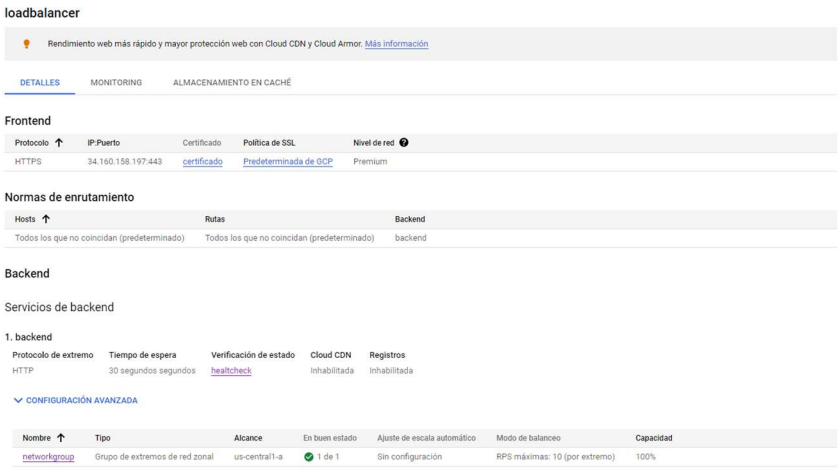
Creamos el health checker:



Verificamos que esta creado correctamente



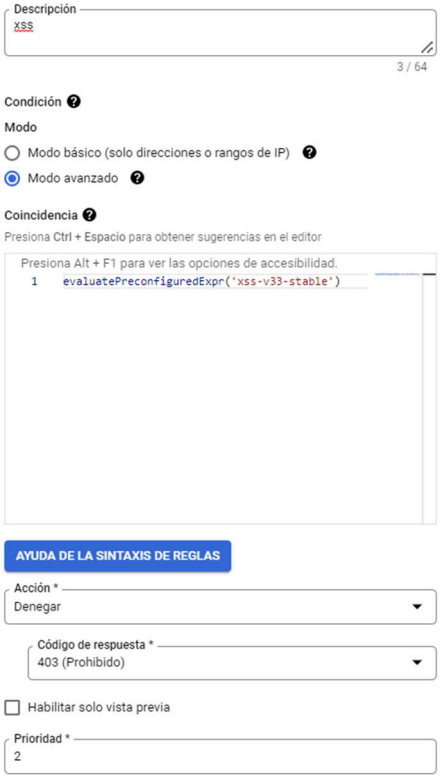
Nombre	Tipo de balanceador de cargas	Protocolos	Región	Backends
frontend-redirect	HTTP(S)	HTTP		✓
loadbalancer	HTTP(S)	HTTPS		✓ 1 servicio de backend (0 grupos de instancias, 1 grupo de extremos de red)



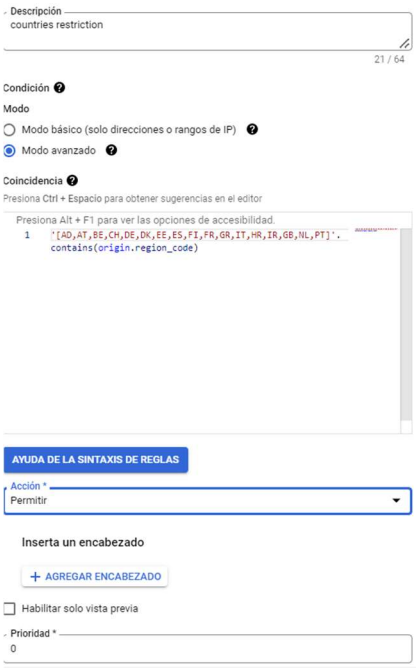
A continuación, queremos establecer unas normas de seguridad para nuestro servidor web, tanto de SQL Injection, Cross Syte Scripting y restricción de tráfico a países de confinaza de la Unión europea. Para ello accedemos a Cloud Armor de Google.

En primer lugar, configuramos las reglas para evitar SQL Injection.

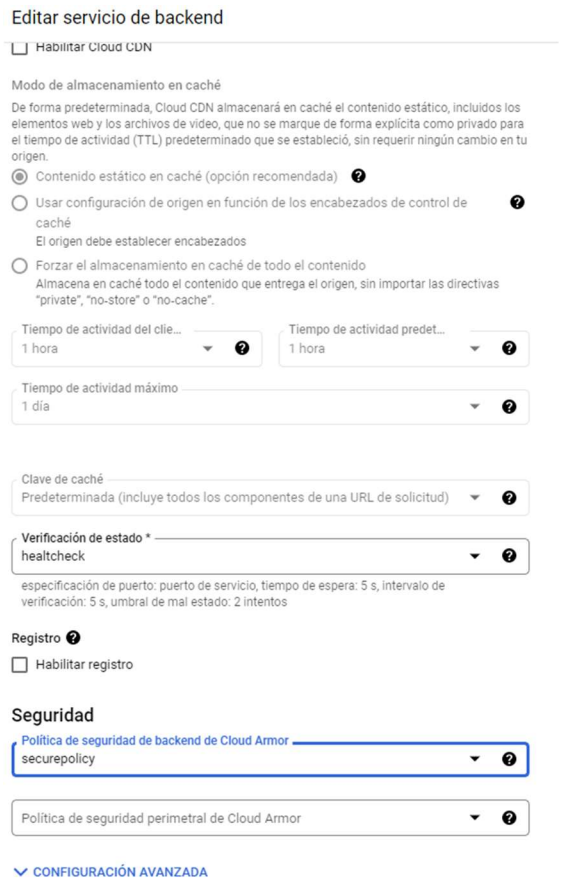
A continuación, configuramos la regla para evitar Cross Syte Scripting.



Por último, restringimos el tráfico a los países amigos de la Unión Europea.



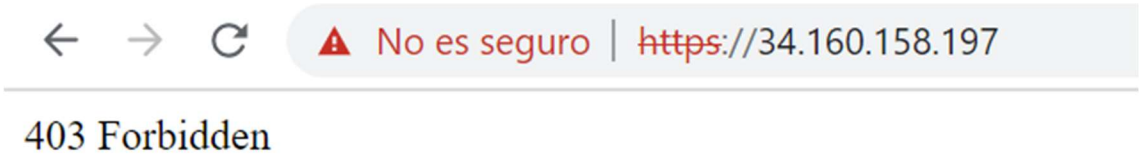
Una vez tenemos la política de seguridad creada, lo añadimos a nuestro backend del load balancer.



Para demostrar que funciona correctamente, nos conectamos a través de nuestro navegador a la dirección pública de nuestro load balancer redirect. De esta forma se puede ver lo siguiente:

Pone que no es seguro porque la certificación está hecha por nosotros mismos, es decir, lo hemos auto firmado y no es una autoridad reconocida por nuestro navegador.

Para comprobar que las reglas funcionan correctamente vamos a eliminar de la lista de países permitidos a España, de esta manera no podremos conectarnos a nuestro servicio web.



¿Qué ventajas e inconvenientes tiene hacer http offloading en el balanceador?
La principal ventaja que tiene utilizar SSL offloading es que el servidor no necesita descifrar y cifrar todos los datos entrantes y salientes, esto permite reducir la carga de trabajo, es decir, nos permite mejorar la velocidad de nuestros servidores.

La desventaja principal es que el tráfico entre el load balancer y el servidor no está cifrado y por lo tanto es vulnerable al robo de datos, al secuestro de sesiones e incluso ataques Man in the middle y además compartir la clave privada del servidor con el load balancer puede ser arriesgado.

¿Qué pasos adicionales has tenido qué hacer para que la máquina pueda salir a internet para poder instalar el servidor nginx?
Al eliminar la ip pública de nuestro servidor web no tenemos acceso a Internet y por tanto no podemos instalar nginx. Para solucionar este problema y como queremos evitar otorgar una ip pública a dicho servidor, implementamos un mecanismo de NAT, un mecanismo utilizado por routers IP para cambiar paquetes entre dos redes que asignan redes incompatibles.
Una vez implementado el mecanismo ya tenemos la posibilidad de conectarnos a la red, sin haber otorgado una dirección ip pública, y de este modo poder descargar nginx.

4ª SOLUCIÓN

¿Qué otras mejoras se te ocurrirían para mejorar la seguridad o disponibilidad del servidor web?

Para mejorar la seguridad podríamos mejorar la funcionalidad de nuestro servicio de seguridad, por ejemplo, implementando nuevas reglas WAF a nuestro cloud armour. En este caso se sugiere la implementación de una regla que evite la detección de escáner. Este ataque, también conocido como port scan consiste en analizar de forma automática todos los puertos de un equipo que esté conectado a la red buscando posibles puertos que estén abiertos o con protocolos de seguridad deficientes para llevar a cabo sus ataques.

Otra posibilidad sería la de implementar protección frente a la inclusión de archivos remotos. Este ataque consiste en incluir archivos maliciosos externos que posteriormente son ejecutados por la web o por una aplicación.

Para mejorar la disponibilidad de nuestro servidor web sería una buena idea levantar el servidor en una zona y los back-up de nuestro servidor en zonas distintas, de este modo si ocurre un problema en una zona determinada podríamos levantar el servidor preparado como back-up y continuar con el servicio.